

Efficiency through reuse in algebraic multigrid

Andrey Prokopenko

Paul Lin

John Shadid

Jonathan Hu

CCR, Sandia National Labs

April 13, 2016

SAND2016-2612 C



Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXX



Motivation

- Multigrid setup times growing with number of MPI ranks
- Many simulations only change the values of a matrix, not the structure
- Workaround for scalability issues

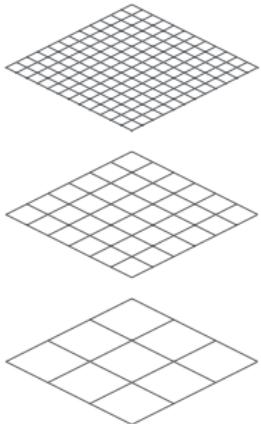
Motivation

- Multigrid setup times growing with number of MPI ranks
- Many simulations only change the values of a matrix, not the structure
- Workaround for scalability issues

Few years ago...

MPI ranks	DOFs	AMG setup time (s)
128	0.8M	9.5
1024	6.5M	10.8
8192	51.0M	12.1
65536	401M	25.5
524288	3.2B	1312

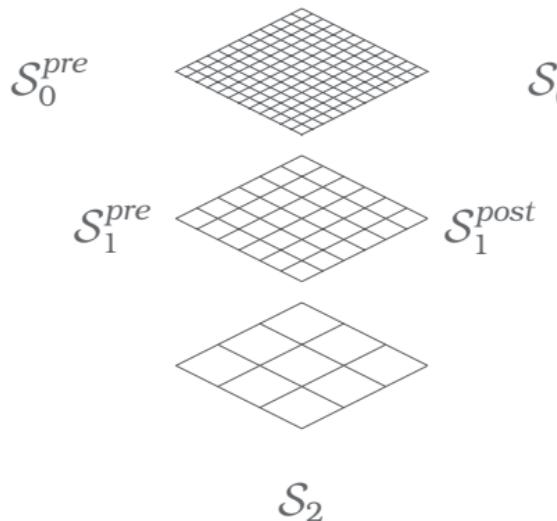
Algebraic Multigrid (AMG)



Main idea

Capture errors at multiple resolutions.

Algebraic Multigrid (AMG)



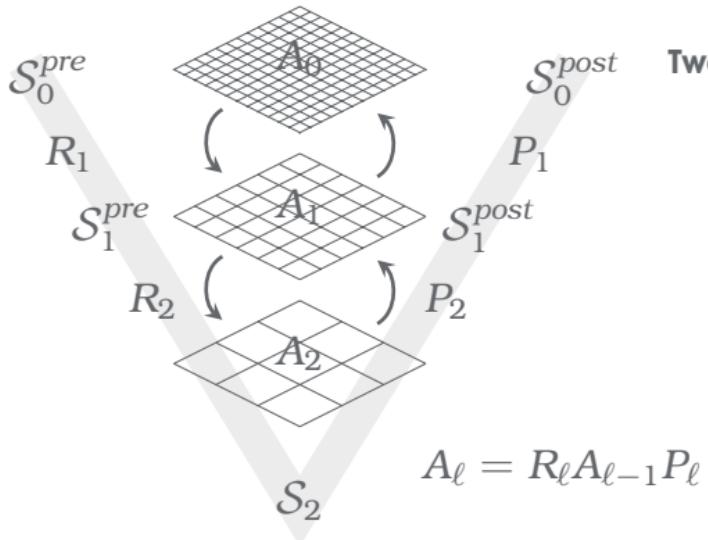
Two main components

- Smoothers
 - Approximate solve on each level
 - “Cheap” reduction of oscillatory error (high energy)
 - $\mathcal{S}_L \approx A_L^{-1}$ on the coarsest level L

Main idea

Capture errors at multiple resolutions.

Algebraic Multigrid (AMG)



Two main components

- Smoothers
 - Approximate solve on each level
 - “Cheap” reduction of oscillatory error (high energy)
 - $S_L \approx A_L^{-1}$ on the coarsest level L
- Grid transfers (prolongators and restrictors)
 - Data movement between levels
 - Reduction of smooth error (low energy)

Main idea

Capture errors at multiple resolutions.

MueLu: Trilinos multigrid library

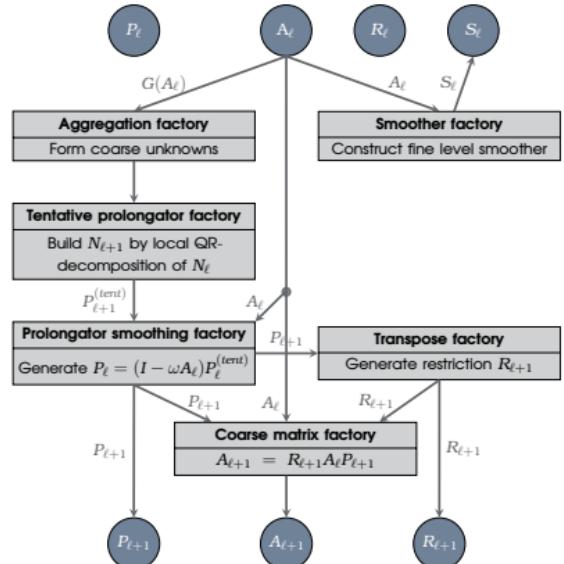
- Can use either EPETRA (32-bit) or TPETRA
Template types: Local and global indices, scalar, compute node
- Grid transfers
 - Smoothed and unsmoothed aggregation
 - Petrov-Galerkin
 - Energy minimization
 - Maxwell
- Smoothers (IFPACK/IFPACK2)
 - Relaxation: Jacobi, SOR, ℓ_1 Gauss-Seidel
 - Incomplete factorizations: ILU(k), ILUT, ILUTP*
 - Others: Chebyshev, additive Schwarz, Krylov, Vanka, ...
- Direct solvers (AMESOS/AMESOS2)
KLU2, SuperLU, ...
- Load balancing (ZOLTAN/ZOLTAN2)
RCB, multijagged (ZOLTAN2 only)

Smoothed Aggregation (SA) main kernels

- Aggregation (forming coarse unknowns)
- Tentative prolongator construction $P_\ell^{(tent)}$
- Smoothed prolongator construction

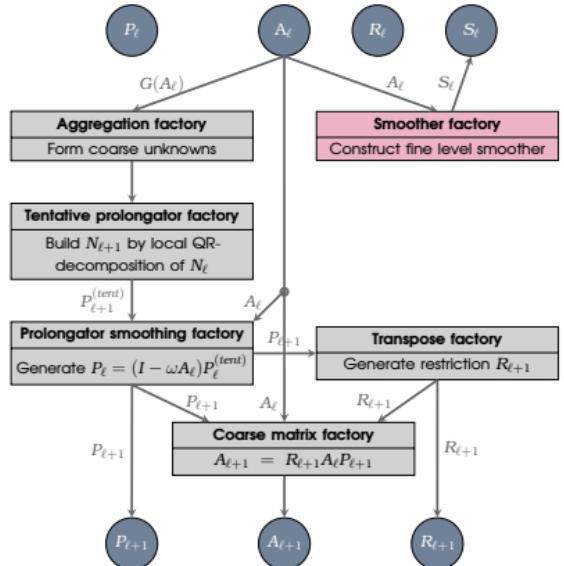
$$P_\ell = (I - \omega D^{-1} A_{\ell-1}) P_\ell^{(tent)}$$
- Coarse matrix construction (matrix-matrix multiply)

$$A_\ell = R_\ell A_{\ell-1} P_\ell$$
- Load balancing of A_ℓ (if necessary)
- Smoother initialization



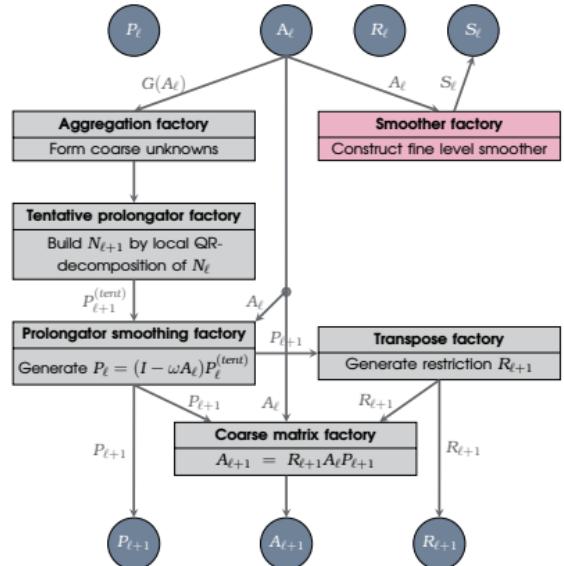
Reuse: fine level smoothers (S)

- Reuse
 - Symbolic factorization of smoother S_0
 - Recompute
 - Smoother S_0 (only numeric factorization)
 - Everything else...
- ✓ Useful for heavy smoothers, like ILU
- ✓ Does not affect convergence
- ✗ Little benefit for light-weight smoothers



Reuse: fine level smoothers (S)

- Additive Schwarz/subdomain ILU
 - Data import infrastructure
 - Local symbolic factorizations
 - Data transfers unavoidable
- Polynomial smoothers
 - Reuse eigenvalue estimate
 - Reuse initial guess for eigenvalue estimate
- Jacobi, Gauss-Seidel, etc.
 - No reuse



Reuse: tentative prolongators (TP)

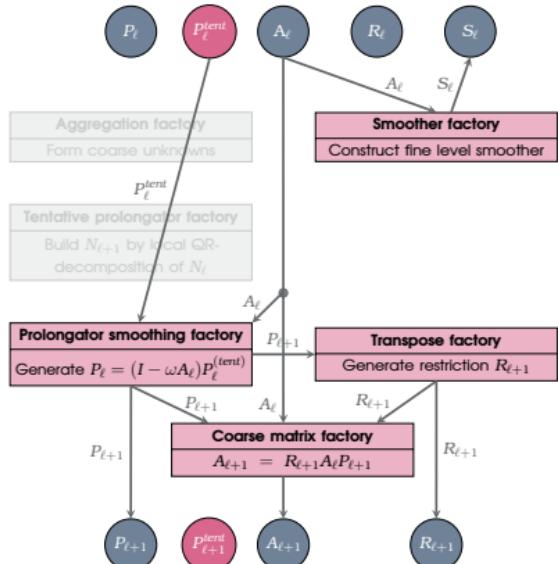
- Reuse

- Tentative prolongators $P_\ell^{(tent)}, \ell > 0$
- Matrix graphs of P_ℓ and $A_\ell, \ell > 0$
- Symbolic factorization of smoothers $S_\ell, \ell \geq 0$

- Recompute

- Smoothed prolongators $P_\ell, \ell > 0$
(reusing matrix graphs)
- Coarse level matrices $A_\ell, \ell > 0$
(reusing matrix graphs)
- Smoothers S_ℓ (only numeric factorization)

- ✓ Avoids construction of tentative prolongator
- ✓ Preserves import objects for rebalancing
- ✓ Does not affect convergence
- ✗ Requires matrix-matrix product for P and RAP



Reuse: final prolongators (RP)

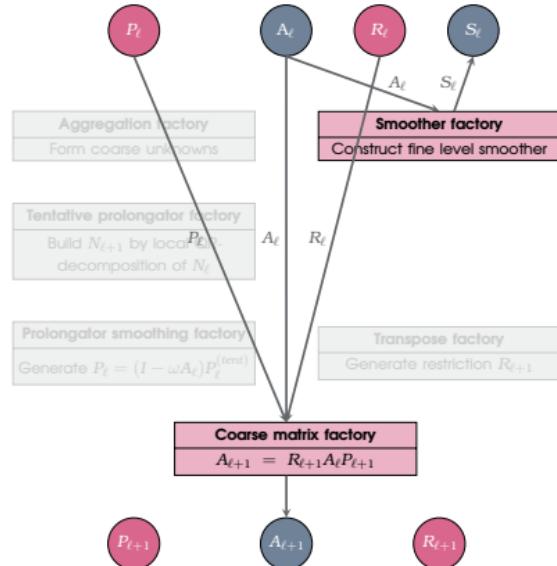
- Reuse

- Prolongators/restrictors $P_\ell, R_\ell, \ell > 0$
- Matrix graphs of $A_\ell, \ell > 0$
- Symbolic factorization of smoothers $S_\ell, \ell \geq 0$

- Recompute

- Coarse level matrices $A_\ell, \ell > 0$ (reusing matrix graphs)
- Smoothers S_ℓ (only numeric factorization)

- ✓ Avoids matrix-matrix product for final P_ℓ
- ✓ Preserves import object for rebalancing A_ℓ , $\ell > 0$
- ✗ May negatively affect convergence
- ✗ Requires matrix-matrix product for RAP



Reuse: all but fine level smoother (RAP)

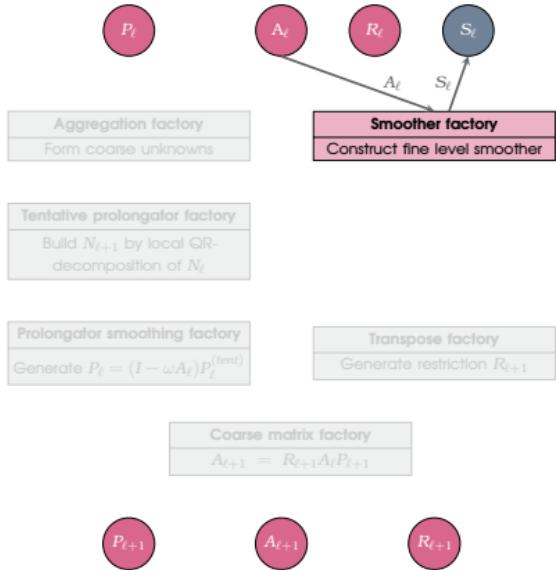
- Reuse

- Prolongators/restrictors $P_\ell, R_\ell, \ell > 0$
- Coarse level matrices $A_\ell, \ell > 0$
- Smoothers $S_\ell, \ell > 0$
- Symbolic factorization of smoothers S_0

- Recompute

- Smoother S_0 (only numeric factorization)

- ✓ No matrix-matrix products required
- ✓ Preserves coarse smoother data
- ✓ Preserves rebalancing information
- ✓ Cheapest reuse option
- ✗ Least likely to converge



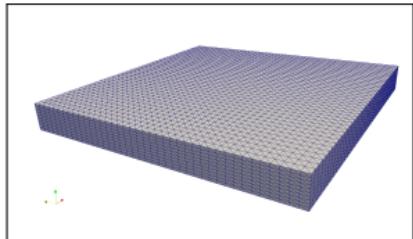
Experiments: ISMIP-HOM Test C

A “first order” approximation to a full Stokes model for a standard ice sheet model benchmark ISMIP-HOM (Test C). This model is an approximation to viscous incompressible quasi-static Stokes flow with power-law viscosity

$$\begin{aligned}-\nabla \cdot (2\mu \dot{\epsilon}_1) &= -\rho g \frac{\partial s}{\partial x} \\-\nabla \cdot (2\mu \dot{\epsilon}_2) &= -\rho g \frac{\partial s}{\partial y}\end{aligned}$$

where

$$\begin{aligned}\dot{\epsilon}_1^T &= (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13}) \\ \dot{\epsilon}_2^T &= (\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23}) \\ \dot{\epsilon}_{ij} &= \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)\end{aligned}$$

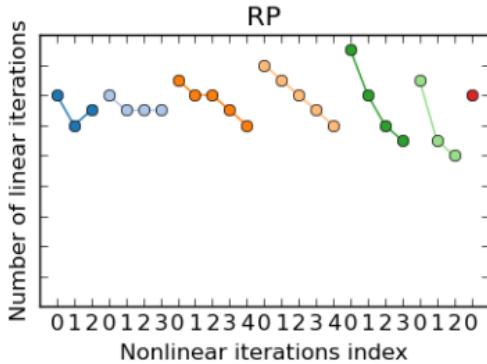
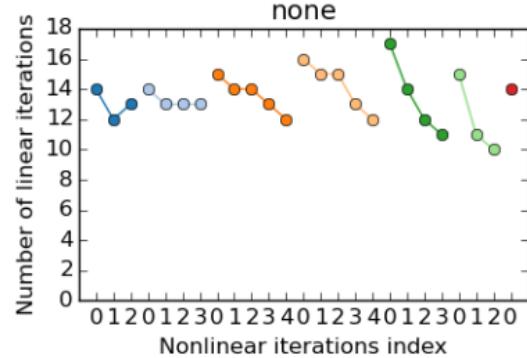


and viscosity μ is a nonlinear function given by “Glen’s law”

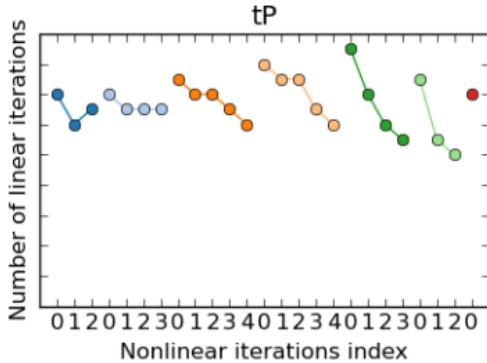
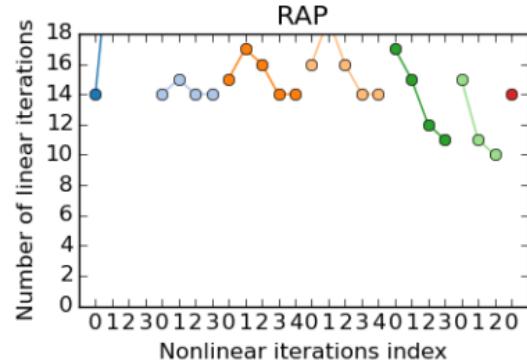
$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \left(\frac{1}{2} \sum_{ij} \dot{\epsilon}_{ij}^2 \right)^{\frac{1}{2n} - \frac{1}{2}}$$

The model is complimented by relevant stress-free and floating ice boundary

Experiments: ISMIP-HOM Test C

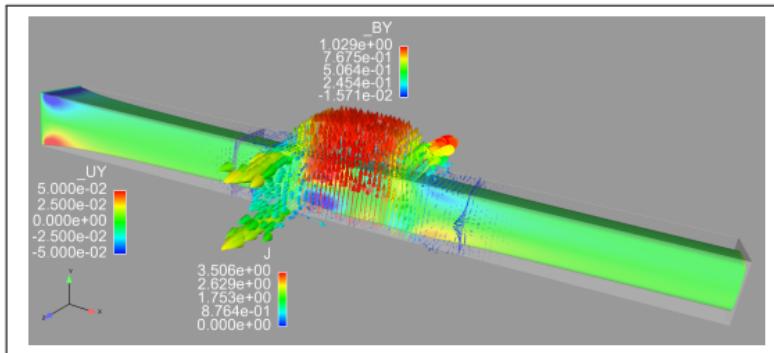


Reuse	Setup	Solve
No reuse	77.2	66.7
TP	64.8	67.4
RP	42.6	67.5
RAP	22.0	84.8



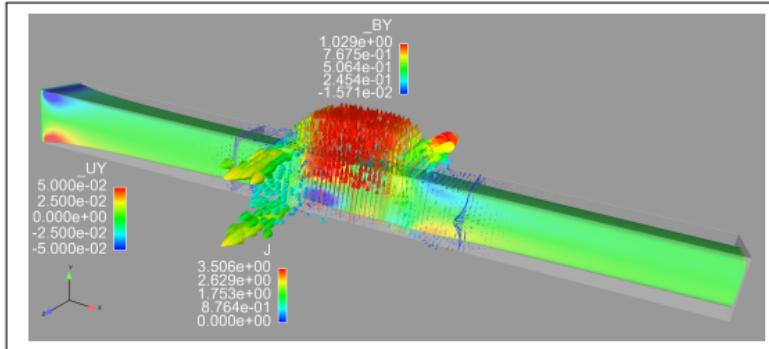
Experiments: 3D MHD generator

- Steady-state 3D MHD generator
 - Resistive MHD model
 - Stabilized FE
 - Newton-Krylov solve
 - 8 DOFs/mesh node
- Monolithic preconditioner
- Prolongator is unsmoothed
- Heavy smoother
 - In this case, DD/ILU(0)
- P cheap to construct compared to smoothers



Experiments: 3D MHD generator

- Steady-state 3D MHD generator
 - Resistive MHD model
 - Stabilized FE
 - Newton-Krylov solve
 - 8 DOFs/mesh node
- Monolithic preconditioner
- Prolongator is unsmoothed
- Heavy smoother
 - In this case, DD/ILU(0)
- P cheap to construct compared to smoothers



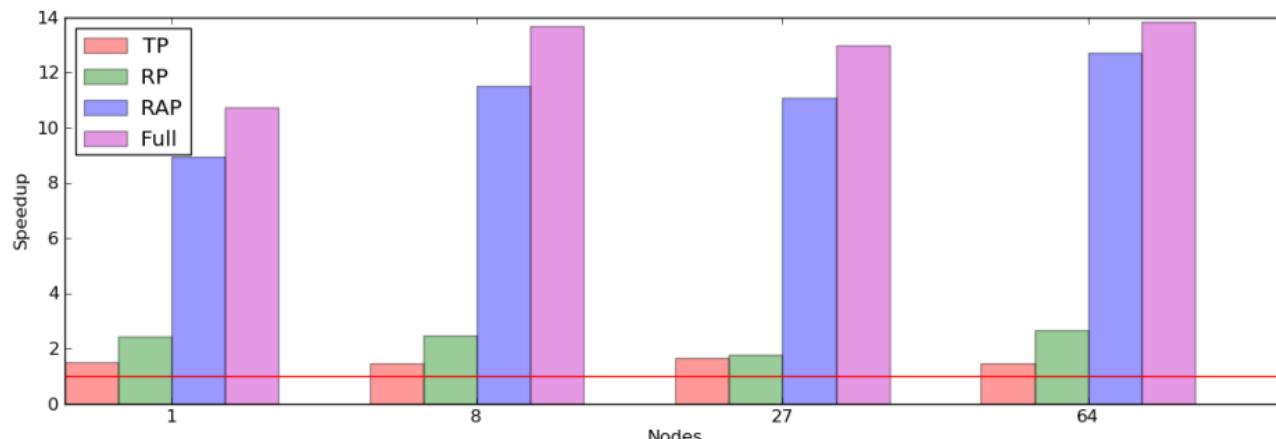
Reuse	Setup	Solve	Total
No reuse	3.79	2.65	6.44
S	3.27	2.63	5.91
RP	3.14	2.61	5.75
RAP	2.74	42.80	45.53

Experiments: jet problem

- 3D Jet, $\text{Re}=106$, CFL 0.25, no slip BCs
- SA AMG, V(3,3) symmetric Gauss-Seidel smoothing
- Setup cost almost entirely
 - Smoothed prolongator
 - $P = (I - \omega D^{-1}A)P^{(tent)}$
 - Galerkin product
- For this particular problem, convergence maintained

Experiments: jet problem

- 3D Jet, $Re=106$, CFL 0.25, no slip BCs
- SA AMG, V(3,3) symmetric Gauss-Seidel smoothing
- Setup cost almost entirely
 - Smoothed prolongator
 - $P = (I - \omega D^{-1}A)P^{(tent)}$
 - Galerkin product



Conclusions

- Easy way of reducing multigrid setup times without changing algorithms
- Multiple opportunities for reducing cost through reuse
 - Grid transfers
 - Heavy weight smoothers
 - Matrix-matrix multiplication
- Effectiveness of reusing setup information is problem dependent
- Still have some low-hanging fruit to pick

Future work

- Storing more auxiliary data
 - Temporary communication data structures
 - Hash tables for matrix-matrix multiplication
- Experimenting with reuse with next-gen smoothers
 - Multi-colored Gauss-Seidel
 - Iterative ILU
- Ability to use different reuse strategies in Newton solver and across transient steps
 - Heavy reuse within Newton solver, lighter reuse across time steps
 - Lighter reuse early in simulation due to startup conditions
- Better information exchange between nonlinear and linear solve

MueLu: references

- A. Prokopenko, J. J. Hu, T. A. Wiesner, C. M. Siefert, and R. S. Tuminaro, MueLu User's Guide 1.0, 2014. SAND2014-18874, Sandia National Laboratories.
- T.A. Wiesner, M.W. Gee, A. Prokopenko, and J.J. Hu, The MueLu tutorial, <http://trilinos.org/packages/muelu/muelu-tutorial>, 2014. SAND2014-18624R, Sandia National Laboratories.
- P. Lin, M. Bettencourt, S. Domino, T. Fisher, M. Hoemmen, J.J. Hu, E. Phipps, A. Prokopenko, S. Rajamanickam, C. Siefert, S. Kennon, Towards extreme-scale simulations for low Mach fluids with second-generation Trilinos, Parallel Processing Letters, World Scientific, 2014, 24, 1442005